

Avoiding IP Fragmentation in GRE Tunnel Deployments

Configuration Guide

Author: Donovan Williams
Consulting Security Engineer

Contents

Introduction	3
Network Components	3
IP Fragmentation and Reassembly Overview	3
TCP Maximum Segment Size (MSS) Overview	3
GRE (Generic Route Encapsulation) Overview	3
Network Architecture	4
FGT-1000C Configuration	5
FGT-3600C Configuration	6
Fortinet TCP-MSS-Sender Option	7
Updated Firewall Policies on the 1000C and 3600C	7
Fortigate 1000C Firewall Policy	8
Fortigate 3600C Firewall Policy	8
BreakingPoint Testing (Clients connecting to servers and downloading 32K files).....	9
First Test	9
Second Test	10

Change Log

Revision	Date	Change Description	Owner
1	2014-06-10	Initial Release	Donovan Williams

Introduction

The purpose of this document is to explain how to avoid IP Fragmentation with the FortiGate TCP Maximum Segment Size feature when deploying FortiGate firewalls in GRE Tunnel mode.

Network Components

The following products were used:

- FortiGate 3600C FG3K6C-5.00-FW-build271
- FortiGate 1000C FGT1KC-4.00-FW-build672
- IXIA Breaking Point version 3.1 emulating clients and servers

IP Fragmentation and Reassembly Overview

IP (Internet Protocol) is used over a wide variety of transmission links. While the maximum length of an IP datagram is 64K Bytes, most transmission links enforce a smaller maximum packet length to accommodate the transmission link. This is called the Path MTU (Maximum Transmission Unit). IP allows network devices such as routers and firewalls to fragment packets in order to accommodate the respective MTU differences.

The receiving host is responsible for reassembling any fragments back to the original IP datagram. IP fragmentation involves breaking a datagram into a number of pieces that can be reassembled later. The IP source, destination, identification, total length, and fragment offset fields, along with the "more fragments" and "don't fragment" flags in the IP header, are used for IP fragmentation and reassembly. For more information on IP fragmentation and reassembly, please see RFC 791.

TCP Maximum Segment Size (MSS) Overview

The TCP Maximum Segment Size (MSS) defines the maximum amount of data that a host is willing to accept in a single TCP/IP datagram. This TCP/IP datagram may be fragmented at the IP layer. The MSS value is sent as a TCP header option only in TCP SYN segments. Each host of a TCP connection reports its MSS value to the each other. The sending host is required to limit the size of data in a single TCP segment to a value less than or equal to the MSS reported by the receiving host.

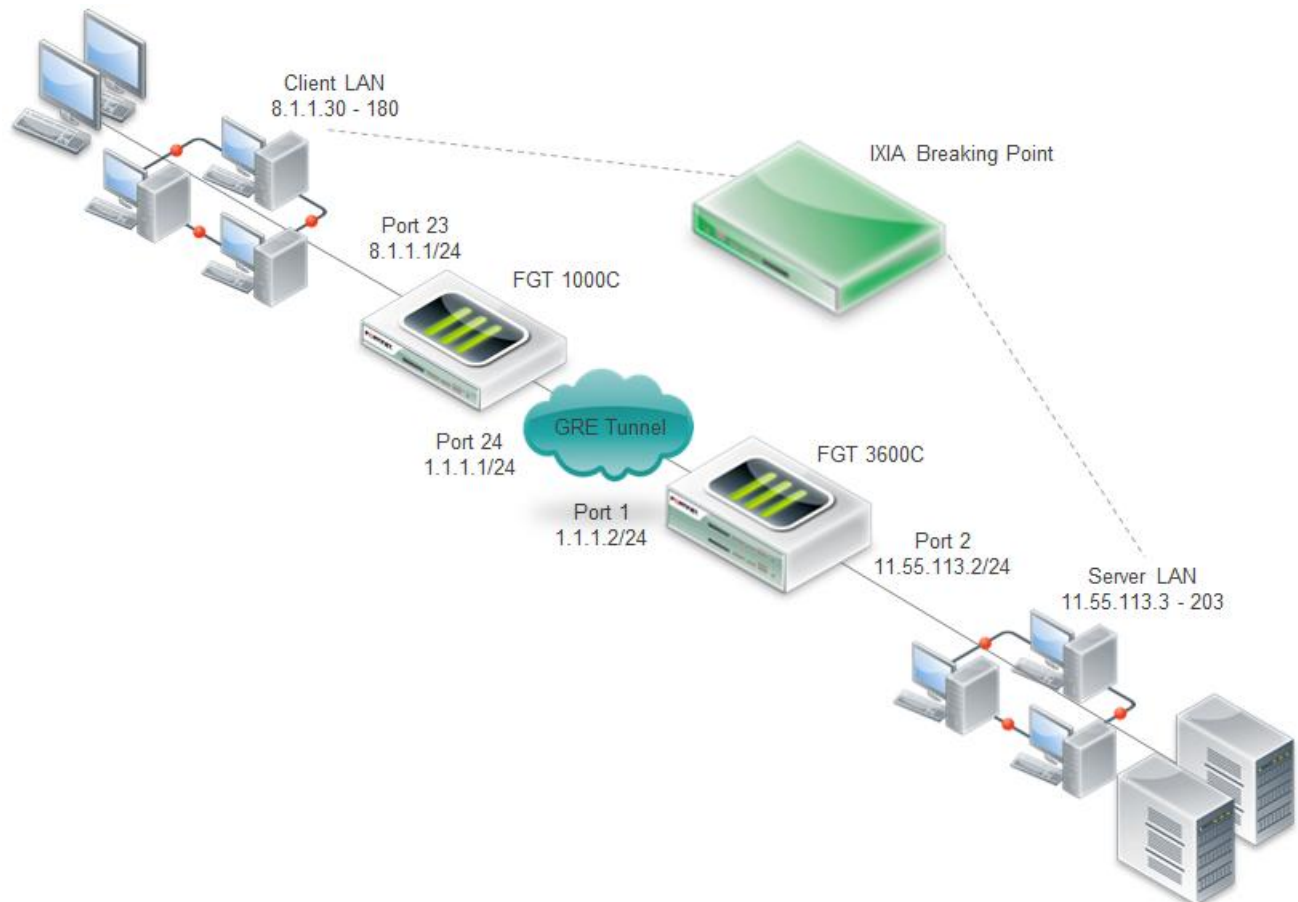
GRE (Generic Route Encapsulation) Overview

Generic Routing Encapsulation (GRE), defined by [RFC 2784](#), is an IP packet encapsulation protocol. A GRE tunnel is used when IP packets need to be sent from one network to another, without being parsed or treated like IP packets by any intervening network devices such as routers or firewalls. GRE encapsulates packets within IP packets and redirects them to an intermediate host (router / firewall), where they are de-encapsulated and routed to their final destination.

Network Architecture

The architecture consists of 150 clients connected to port 23 on the FortiGate 1000C. The clients are part of subnet 8.1.1.0. A GRE tunnel has been implemented between port 24 on the FortiGate 1000C and port 1 of the FortiGate 3600C. The Servers are connected to port 2 of the FortiGate 3600C and are on subnet 11.55.113.0. Each client is running an HTTP 1.1 browser and downloading a 32K file from the servers. An Ixia Breaking Point was used to emulate the clients connecting to servers.

Figure 1 – Network Diagram



FGT-1000C Configuration

```
#config-version=FGT1KC-4.00-FW-build672-130904:opmode=0:vdom=0:user=admin
#global_vdom=1
config system global
    set hostname "LAB16-FG1000C-01"
end
config system interface
    edit "mgmt1"
        set vdom "root"
        set ip 10.25.16.1 255.255.0.0
        set allowaccess ping https ssh fgfm
    next
    edit "port23"
        set vdom "root"
        set ip 8.1.1.1 255.255.255.0
        set allowaccess ping
        set alias "inside "
    next
    edit "port24"
        set vdom "root"
        set ip 1.1.1.1 255.255.255.0
        set alias "to3600P1"
    next
    edit "to3600C"
        set vdom "root"
        set type tunnel
        set interface "port24"
    next
end

config firewall policy
    edit 1
        set srcintf "port23"
        set dstintf "to3600C"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ANY"

    next
    edit 2
        set srcintf "to3600C"
        set dstintf "port23"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ANY"
    next
end

config router static
    edit 1
        set device "to3600C"
        set dst 11.55.113.0 255.255.255.0
    next
end
```

FGT-3600C Configuration

```
#config-version=FG3K6C-5.00-FW-build271-140124:opmode=0:vdom=0:user=admin
#global_vdom=1
config system global
    set hostname "LAB01-FG3600C-01"
end
config system interface
    edit "port1"
        set vdom "root"
        set ip 1.1.1.2 255.255.255.0
        set allowaccess ping fgfm
        next
    edit "port2"
        set vdom "root"
        set ip 11.55.113.2 255.255.255.0
        set allowaccess ping
        set alias "Outside Servers "
        next
    edit "mgmt"
        set vdom "root"
        set ip 10.55.100.111 255.255.252.0
        set allowaccess ping https fgfm
        set dedicated-to management
        next
    edit "to1000C"
        set vdom "root"
        set type tunnel
        set snmp-index 40
        set interface "port1"
    next
end
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "to1000C"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "to1000C"
        set dstintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
config router static
    edit 1
        set device "to1000C"
        set dst 8.1.1.0 255.255.255.0
    next
end
```

For detailed information on configuring GRE tunnels with static routes, please refer to the Fortinet Knowledge Base Technical Note:

<http://kb.fortinet.com/kb/microsites/search.do?cmd=displayKC&docType=kc&externalId=FD31182>

Fortinet TCP-MSS-Sender Option

In the diagram the clients and servers receive an MTU from their connected Ethernet interface and then calculate the MSS value ($1500 - 40 = 1460$).

The MTU of Ethernet is 1500. The MSS number is 40 bytes smaller than the MTU because the MSS value is the TCP data size. The 20 byte IP header and 20 byte TCP header are subtracted leaving the value 1460 as the value the clients and servers send to each other as the negotiated MSS. GRE encapsulation adds an additional 24 bytes to the original IP packet (4 byte GRE header + 20 byte IP header). The clients and servers are not aware of the GRE tunnel in the path and as a result, data communications will fail due to fragmentation. The clients and servers should calculate an MSS value of 1436, ($1500 - 40 - 24$) to accommodate the MTU of the GRE tunnel.

The following option needs to be added to the firewall policies to set the MSS value to 1436.

<code>tcp-mss-sender</code> <code><maximumsize_int></code>	<p>Enter a TCP Maximum Sending Size number for the sender.</p> <p>When a FortiGate unit is configured to use PPPoE to connect to an ISP, certain web sites may not be accessible to users. This occurs because a PPPoE frame takes an extra 8 bytes off the standard Ethernet MTU of 1500.</p> <p>When the server sends the large packet with DF bit set to 1, the ADSL provider's router either does not send an "ICMP fragmentation needed" packet or the packet is dropped along the path to the web server. In either case, the web server never knows fragmentation is required to reach the client.</p> <p>In this case, configure the <code>tcp-mss-sender</code> option to enable access to all web sites. For more information, see the article Cannot view some web sites when using PPPoE on the Fortinet Knowledge Center.</p>	0
---	--	---

Updated Firewall Policies on the 1000C and 3600C

Fortigate 1000C Firewall Policy

```
config firewall policy
edit 1
    set srcintf "port23"
    set dstintf "to3600C"
        set srcaddr "all"
        set dstaddr "all"
    set action accept
    set schedule "always"
    set service "ANY"
    set tcp-mss-sender 1436
next
edit 2
    set srcintf "to3600C"
    set dstintf "port23"
        set srcaddr "all"
        set dstaddr "all"
    set action accept
    set schedule "always"
    set service "ANY"
next
end
```

Fortigate 3600C Firewall Policy

```
config firewall policy
edit 1
    set srcintf "port2"
    set dstintf "to1000C"
    set srcaddr "all"
    set dstaddr "all"
    set action accept
    set schedule "always"
    set service "ALL"
    set tcp-mss-sender 1436
next
edit 2
    set srcintf "to1000C"
    set dstintf "port2"
    set srcaddr "all"
    set dstaddr "all"
    set action accept
    set schedule "always"
    set service "ALL"
next
end
```


BreakingPoint Testing (Clients connecting to servers and downloading 32K files)

In this scenario, two tests were run:

- The first test consists of clients connecting to servers and the tcp-mss-sender values are NOT configured.
- The second test repeats the first, but with the corrected tcp-mss-sender values configured.

First Test

While the clients are connecting and trying to download the files, a “diagnose debug packet” was implemented on the client facing interface of the FortiGate 1000C (shown on the left) and the server facing interface on the FortiGate 3600C (shown on the right).

```

1. telnet
New Info Close Execute Profiles
telnet
14.571270 8.1.1.79.39720 -> 11.55.113.129.80: psh 327550363 ack 2478304178
14.571768 11.55.113.129.80 -> 8.1.1.79.39720: ack 327550530
14.580369 8.1.1.88.45951 -> 11.55.113.21.80: syn 2490633378
14.590308 arp who-has 8.1.1.88 tell 8.1.1.1
14.590444 arp reply 8.1.1.88 is-at 2:1a:c1:1:0:58
14.590445 8.1.1.59.46622 -> 11.55.113.167.80: syn 347204180
14.590447 11.55.113.21.80 -> 8.1.1.88.45951: syn 340403041 ack 2490633379
14.590693 8.1.1.88.45951 -> 11.55.113.21.80: ack 340403042
14.591182 8.1.1.88.45951 -> 11.55.113.21.80: psh 2490633379 ack 340403042
14.591923 11.55.113.21.80 -> 8.1.1.88.45951: ack 2490633533
14.600277 8.1.1.167.27050 -> 11.55.113.132.80: syn 354145490
14.600308 arp who-has 8.1.1.59 tell 8.1.1.1
14.600521 arp reply 8.1.1.59 is-at 2:1a:c1:1:0:3b
14.600523 11.55.113.167.80 -> 8.1.1.59.46622: syn 2492847498 ack 347204181
14.600767 8.1.1.59.46622 -> 11.55.113.167.80: ack 2492847499
14.600776 11.55.113.132.80 -> 8.1.1.167.27050: syn 2503910027 ack 354145491
14.601816 8.1.1.167.27050 -> 11.55.113.132.80: ack 2503910028
14.601258 8.1.1.59.46622 -> 11.55.113.167.80: psh 347204181 ack 2492847499
14.601506 8.1.1.167.27050 -> 11.55.113.132.80: psh 354145491 ack 2503910028
14.601755 11.55.113.167.80 -> 8.1.1.59.46622: ack 347204181
14.602246 11.55.113.132.80 -> 8.1.1.167.27050: ack 354145491
14.610352 8.1.1.47.53546 -> 11.55.113.147.80: syn 361159096
14.620308 arp who-has 8.1.1.47 tell 8.1.1.1
14.620430 arp reply 8.1.1.47 is-at 2:1a:c1:1:0:2f
14.620430 8.1.1.153.24397 -> 11.55.113.8.80: syn 2513550814
14.620432 11.55.113.147.80 -> 8.1.1.47.53546: syn 361379281 ack 361159097
14.620926 8.1.1.47.53546 -> 11.55.113.147.80: ack 361379282
14.621413 8.1.1.47.53546 -> 11.55.113.147.80: psh 361159097 ack 361379282
14.621910 11.55.113.147.80 -> 8.1.1.47.53546: ack 361159251
14.630362 8.1.1.89.56234 -> 11.55.113.134.80: syn 375214207
14.630368 arp who-has 8.1.1.153 tell 8.1.1.1
14.630508 arp reply 8.1.1.153 is-at 2:1a:c1:1:0:99
14.630511 11.55.113.8.80 -> 8.1.1.153.24397: syn 2518902354 ack 2513550815
14.630760 8.1.1.153.24397 -> 11.55.113.8.80: ack 2518902355
14.631245 8.1.1.153.24397 -> 11.55.113.8.80: psh 2513550815 ack :1a:c1:1:0:74
14.670578 11.55.113.39.80 -> 8.1.1.116.60062: syn 402709958 ack 2552935491
2660 packets received by filter
0 packets dropped by kernel
LAB16-FG1000C-01 #

14.656850 8.1.1.141.1436 -> 11.55.113.36.80: psh 1775557997 ack 4270666092
14.657097 11.55.113.36.80 -> 8.1.1.141.1436: ack 1775558151
14.657588 11.55.113.36.80 -> 8.1.1.141.1436: psh 4270666092 ack 1775558151
14.657998 11.55.113.36.80 -> 8.1.1.141.1436: psh 4270666092 ack 1775558151
14.657615 11.55.113.2 -> 11.55.113.36: icmp: 8.1.1.141 unreachable - need to frag (mtu 1476)
14.657633 11.55.113.2 -> 11.55.113.36: icmp: 8.1.1.141 unreachable - need to frag (mtu 1476)
15.155837 8.1.1.79.39720 -> 11.55.113.129.80: syn 327550362
15.156062 11.55.113.129.80 -> 8.1.1.79.39720: syn 2478304177 ack 327550363
15.156554 11.55.113.36.80 -> 8.1.1.141.1436: psh 4270666092 ack 1775558151
15.156563 11.55.113.36.80 -> 8.1.1.141.1436: psh 4270666092 ack 1775558151
15.156570 11.55.113.2 -> 11.55.113.36: icmp: 8.1.1.141 unreachable - need to frag (mtu 1476)
15.156574 11.55.113.2 -> 11.55.113.36: icmp: 8.1.1.141 unreachable - need to frag (mtu 1476)
15.164188 8.1.1.79.39720 -> 11.55.113.129.80: ack 2478304178
15.164663 8.1.1.79.39720 -> 11.55.113.129.80: psh 327550363 ack 2478304178
15.164909 11.55.113.129.80 -> 8.1.1.79.39720: ack 327550530
15.165399 11.55.113.129.80 -> 8.1.1.79.39720: psh 2478304178 ack 327550530
15.165409 11.55.113.129.80 -> 8.1.1.79.39720: psh 2478305638 ack 327550530
15.165424 11.55.113.2 -> 11.55.113.129: icmp: 8.1.1.79 unreachable - need to frag (mtu 1476)
15.165442 11.55.113.2 -> 11.55.113.129: icmp: 8.1.1.79 unreachable - need to frag (mtu 1476)
15.173785 8.1.1.88.45951 -> 11.55.113.21.80: syn 2490633378
15.174004 11.55.113.21.80 -> 8.1.1.88.45951: syn 340403041 ack 2490633379
15.183875 8.1.1.59.46622 -> 11.55.113.167.80: syn 347204180
15.184080 11.55.113.167.80 -> 8.1.1.59.46622: syn 2492847498 ack 347204181
15.184330 8.1.1.88.45951 -> 11.55.113.21.80: ack 340403042
15.184819 8.1.1.88.45951 -> 11.55.113.21.80: psh 2490633379 ack 340403042
15.185065 11.55.113.21.80 -> 8.1.1.88.45951: ack 2490633533
15.185555 11.55.113.21.80 -> 8.1.1.88.45951: psh 340403042 ack 2490633533
15.185564 11.55.113.21.80 -> 8.1.1.88.45951: psh 340403042 ack 2490633533
15.185574 11.55.113.2 -> 11.55.113.21: icmp: 8.1.1.88 unreachable - need to frag (mtu 1476)
15.185579 11.55.113.2 -> 11.55.113.21: icmp: 8.1.1.88 unreachable - need to frag (mtu 1476)
15.193706 8.1.1.167.27050 -> 11.55.113.132.80: syn 354145490
15.193915 11.55.113.132.80 -> 8.1.1.167.27050: syn 2503910027 ack 354145491
15.194161 8.1.1.59.46622 -> 11.55.113.167.80: ack 2492847499
15.194658 8.1.1.167.27050 -> 11.55.113.132.80: ack 2503910028
15.194673 8.1.1.59.46622 -> 11.55.113.167.80: psh 347204181 ack 2492847499
15.47 unreachable - need to frag (mtu 1476)
15.215583 11.55.113.2 -> 11.55.113.147: icmp: 8.1.1.47 unreachable - need to frag (mtu 1476)
6124 packets received by filter
0 packets dropped by kernel
LAB01-FG3600C-01 #

```

We can observe the SYN packets with some acknowledgments but we also observe “ICMP unreachable” and “need to fragment” messages. Below we can also observe that the connections are not completing since there are no FIN packets and all sessions are being reset.

```

1. telnet

LAB16-FG1000C-01 #
LAB16-FG1000C-01 #
LAB16-FG1000C-01 # diag sniffer packet port23 'tcp[13] & 1 = 1'
interfaces=[port23]
filters=[tcp[13] & 1 = 1]

0 packets received by filter
0 packets dropped by kernel

LAB16-FG1000C-01 # diag sniffer packet port23 'tcp[13] & 4 = 4'
interfaces=[port23]
filters=[tcp[13] & 4 = 4]
0.031039 11.55.113.67.80 -> 8.1.1.35.32296: rst 88486636 ack 2237916181
0.032026 11.55.113.127.80 -> 8.1.1.74.47578: rst 87106263 ack 86927930
0.032515 11.55.113.132.80 -> 8.1.1.139.45412: rst 2233216484 ack 2232975940
0.032520 11.55.113.141.80 -> 8.1.1.149.25986: rst 89211041 ack 2238598449
0.035466 11.55.113.50.80 -> 8.1.1.43.48073: rst 2238771875 ack 2238576842
0.035469 11.55.113.93.80 -> 8.1.1.90.17845: rst 2239473549 ack 2239278610
0.036442 11.55.113.84.80 -> 8.1.1.164.9100: rst 91775614 ack 91296113
0.036685 11.55.113.106.80 -> 8.1.1.75.9234: rst 91785876 ack 2236682626
0.037183 11.55.113.168.80 -> 8.1.1.158.32245: rst 91093720 ack 2235983464
0.037424 11.55.113.8.80 -> 8.1.1.95.24029: rst 2242058616 ack 91113271
0.037427 11.55.113.46.80 -> 8.1.1.57.29212: rst 2242592035 ack 2242106573
0.038155 11.55.113.198.80 -> 8.1.1.131.56700: rst 2236950255 ack 92003303
0.038159 11.55.113.5.80 -> 8.1.1.75.27104: rst 93677291 ack 2243678533
0.038161 11.55.113.114.80 -> 8.1.1.69.60324: rst 2242751052 ack 2237185614
0.038169 11.55.113.197.80 -> 8.1.1.41.13652: rst 92262727 ack 2237241768
0.038648 11.55.113.12.80 -> 8.1.1.143.34729: rst 92961532 ack 2242977754
0.039157 11.55.113.55.80 -> 8.1.1.80.41427: rst 2243927879 ack 93951226
0.039638 11.55.113.159.80 -> 8.1.1.57.27212: rst 2243240539 ack 2243678758
0.041116 11.55.113.98.80 -> 8.1.1.89.11725: rst 2243925923 ack 2243532020

LAB01-FG3600C-01 #
LAB01-FG3600C-01 #
LAB01-FG3600C-01 # diag sniffer packet port2 'tcp[13] & 1 = 1'
interfaces=[port2]
filters=[tcp[13] & 1 = 1]

0 packets received by filter
0 packets dropped by kernel

LAB01-FG3600C-01 # diag sniffer packet port2 'tcp[13] & 4 = 4'
interfaces=[port2]
filters=[tcp[13] & 4 = 4]
0.631613 11.55.113.159.80 -> 8.1.1.57.27212: rst 2243240539 ack 2243678758
0.633072 11.55.113.98.80 -> 8.1.1.89.11725: rst 2243925923 ack 2243532020
0.633335 11.55.113.60.80 -> 8.1.1.78.45713: rst 2246192937 ack 2240677233
0.633561 11.55.113.113.80 -> 8.1.1.113.34406: rst 95080132 ack 2239995609
0.633584 11.55.113.59.80 -> 8.1.1.127.56977: rst 2240452364 ack 2240042876
0.634541 11.55.113.66.80 -> 8.1.1.92.9693: rst 2241798264 ack 2241432058
0.635036 11.55.113.79.80 -> 8.1.1.170.9688: rst 2241754611 ack 96045558
0.636056 11.55.113.40.80 -> 8.1.1.88.25477: rst 2246669231 ack 96750564
0.637501 11.55.113.52.80 -> 8.1.1.115.51272: rst 2247401127 ack 98296384
0.637998 11.55.113.69.80 -> 8.1.1.163.10074: rst 2248060001 ack 98146014
0.638023 11.55.113.25.80 -> 8.1.1.64.3009: rst 2244583779 ack 2244243099
0.638244 11.55.113.200.80 -> 8.1.1.113.2173: rst 97095445 ack 96696321
0.639953 11.55.113.172.80 -> 8.1.1.40.34763: rst 2244862053 ack 2246977892
0.640198 11.55.113.44.80 -> 8.1.1.75.11029: rst 2250316321 ack 2244988927
0.640208 11.55.113.118.80 -> 8.1.1.119.36310: rst 2250252794 ack 2250679405
0.640961 11.55.113.10.80 -> 8.1.1.31.28376: rst 99362127 ack 98097785
0.641224 11.55.113.174.80 -> 8.1.1.156.56018: rst 99145707 ack 2245085838
0.642403 11.55.113.74.80 -> 8.1.1.53.16031: rst 2251479084 ack 2251236997
0.642419 11.55.113.100.80 -> 8.1.1.43.6133: rst 2250327395 ack 100402349

```

Second Test

The “tcp-mss-sender” option is implemented in this test and we can observe that all sessions are completed. FIN packets show sessions closing properly and there are no resets.

```

1. telnet

telnet
New Info Close
Execute
Default Profiles

0.710203 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602364287 ack 456516860
0.710208 11.55.113.46.80 -> 8.1.1.122.45681: syn 2609511782 ack 464358140
0.710211 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602365723 ack 456516860
0.710214 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602367159 ack 456516860
0.710216 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602367207 ack 456516860
0.710219 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602368643 ack 456516860
0.710222 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602370079 ack 456516860
0.710225 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602370127 ack 456516860
0.710228 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602371563 ack 456516860
0.710233 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602372999 ack 456516860
0.710236 11.55.113.3.80 -> 8.1.1.95.47652: ack 450358768
0.710239 11.55.113.184.80 -> 8.1.1.95.47652: ack 450359989
0.710242 11.55.113.38.80 -> 8.1.1.117.2857: psh 462071184 ack 2606102728
0.710244 11.55.113.38.80 -> 8.1.1.117.2857: psh 462072620 ack 2606102728
0.710247 11.55.113.38.80 -> 8.1.1.117.2857: psh 462074056 ack 2606102728
0.710250 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601795485 ack 2604658938
0.710252 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601796921 ack 2604658938
0.710255 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601798357 ack 2604658938
0.710257 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601798405 ack 2604658938
0.710260 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601799041 ack 2604658938
0.710263 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601801277 ack 2604658938
0.710265 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601801325 ack 2604658938
0.710268 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601802761 ack 2604658938
0.710271 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601804197 ack 2604658938
0.710274 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602364322 ack 2606937896
0.710277 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602365758 ack 2606937896
0.710280 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602367194 ack 2606937896
0.710283 11.55.113.17.80 -> 8.1.1.118.45987: fin 2597930873 ack 2597653673
0.710285 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602367242 ack 2606937896
0.710295 8.1.1.31.52234 -> 11.55.113.182.80: ack 2605721898
0.710298 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602368678 ack 2606937896
0.710299 8.1.1.82.23277 -> 11.55.113.27.80: ack 460356401
0.710301 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602370114 ack 2606937896
0.710302 8.1.1.82.23277 -> 11.55.113.27.80: ack 460356401
0.710305 8.1.1.82.23277 -> 11.55.113.27.80: ack 460356401
0.710310316 8.1.1.56.64929 -> 11.55.113.179.80: ack 461076782
0.710316 8.1.1.56.64929 -> 11.55.113.179.80: ack 461076782

2084266 packets received by filter
2076041 packets dropped by kernel

LAB16-FG1000C-01 #

0.194805 11.55.113.57.80 -> 8.1.1.154.13769: psh 2531958573 ack 2531336947
0.194807 11.55.113.188.80 -> 8.1.1.70.23762: ack 2531149313
0.194808 8.1.1.119.31241 -> 11.55.113.6.80: ack 2529568614
0.194809 11.55.113.57.80 -> 8.1.1.154.13769: psh 2531952009 ack 2531336947
0.194811 8.1.1.102.29382 -> 11.55.113.159.80: fin 3768566295 ack 376321843
0.194811 11.55.113.22.80 -> 8.1.1.47.5538: ack 386662349
0.194814 11.55.113.57.80 -> 8.1.1.154.13769: psh 2531952057 ack 2531336947
0.194815 11.55.113.179.80 -> 8.1.1.167.24116: psh 379765924 ack 380356475
0.194818 11.55.113.57.80 -> 8.1.1.154.13769: psh 2531953493 ack 2531336947
0.194820 8.1.1.67.11200 -> 11.55.113.49.80: syn 2537494368
0.194819 11.55.113.179.80 -> 8.1.1.167.24116: psh 379767360 ack 380356475
0.194822 11.55.113.57.80 -> 8.1.1.154.13769: psh 2531954929 ack 2531336947
0.194823 8.1.1.170.55856 -> 11.55.113.105.80: ack 2532847381
0.194826 11.55.113.7.80 -> 8.1.1.166.52100: psh 384055898 ack 382953497
0.194824 11.55.113.179.80 -> 8.1.1.167.24116: psh 379768796 ack 380356475
0.194826 8.1.1.55.30398 -> 11.55.113.62.80: fin 374563212 ack 375919873
0.194830 8.1.1.107.41493 -> 11.55.113.20.80: psh 387357897 ack 386762181
0.194829 11.55.113.7.80 -> 8.1.1.166.52100: psh 384057334 ack 382953497
0.194831 11.55.113.179.80 -> 8.1.1.167.24116: psh 379768844 ack 380356475
0.194833 8.1.1.124.21324 -> 11.55.113.49.80: ack 2528701352
0.194834 11.55.113.7.80 -> 8.1.1.166.52100: psh 3840658770 ack 382953497
0.194835 11.55.113.179.80 -> 8.1.1.167.24116: psh 379770280 ack 380356475
0.194836 8.1.1.170.55856 -> 11.55.113.105.80: ack 2532848865
0.194837 11.55.113.7.80 -> 8.1.1.166.52100: psh 384058818 ack 382953497
0.194839 11.55.113.179.80 -> 8.1.1.167.24116: psh 379771716 ack 380356475
0.194841 11.55.113.7.80 -> 8.1.1.166.52100: psh 384060254 ack 382953497
0.194843 11.55.113.179.80 -> 8.1.1.167.24116: psh 379771764 ack 380356475
0.194844 8.1.1.72.15631 -> 11.55.113.8.80: syn 2537466885
0.194845 11.55.113.7.80 -> 8.1.1.166.52100: psh 384061690 ack 382953497
0.194847 8.1.1.124.21324 -> 11.55.113.49.80: ack 2528702788
0.194847 11.55.113.179.80 -> 8.1.1.167.24116: psh 379773200 ack 380356475
0.194849 11.55.113.7.80 -> 8.1.1.166.52100: psh 384061738 ack 382953497
0.194850 8.1.1.170.55856 -> 11.55.113.105.80: ack 2532848865
0.194851 11.55.113.179.80 -> 8.1.1.167.24116: psh 379774636 ack 380356475
0.194853 11.55.113.7.80 -> 8.1.1.166.52100: psh 384063174 ack 382953497
0.194853 8.1.1.124.21324 -> 11.55.113.49.80: 0.194070 11.55.113.11.80 -> 8.1.1.72.31340: ack 375285210

2305279 packets received by filter
2298265 packets dropped by kernel
.31340: ack 375285210

LAB01-FG3600C-01 #

```

```

1. telnet

LAB16-FG1000C-01 #
LAB16-FG1000C-01 #
LAB16-FG1000C-01 # diag sniffer packet port23 'tcp[13] & 4 = 4'
interfaces=[port23]
filters=[tcp[13] & 4 = 4]

0 packets received by filter
0 packets dropped by kernel

LAB16-FG1000C-01 # diag sniffer packet port23 'tcp[13] & 1 = 1'
interfaces=[port23]
filters=[tcp[13] & 1 = 1]
0.141008 8.1.1.133.42562 -> 11.55.113.19.80: fin 1324944284 ack 3469804094
0.141012 8.1.1.40.6692 -> 11.55.113.3.80: fin 1324944086 ack 1325250354
0.141144 11.55.113.61.80 -> 8.1.1.59.42463: fin 1324480130 ack 1324943566
0.141279 8.1.1.59.42463 -> 11.55.113.61.80: fin 1324943566 ack 1324480131
0.141283 11.55.113.41.80 -> 8.1.1.84.1543: fin 3469848490 ack 1324053430
0.141340 11.55.113.85.80 -> 8.1.1.47.38655: fin 3469749909 ack 1324063537
0.141363 11.55.113.176.80 -> 8.1.1.54.64540: fin 1324372496 ack 3469459378
0.141410 11.55.113.4.80 -> 8.1.1.80.6602: fin 3469707758 ack 1324943224
0.141536 8.1.1.54.64540 -> 11.55.113.176.80: fin 3469459378 ack 1324372497
0.141544 8.1.1.47.38655 -> 11.55.113.85.80: fin 1324063537 ack 3469749910
0.141570 8.1.1.80.6602 -> 11.55.113.4.80: fin 1324943224 ack 3469707759
0.141598 11.55.113.111.80 -> 8.1.1.50.8483: fin 1325814853 ack 1325501489
0.141717 8.1.1.84.1543 -> 11.55.113.41.80: fin 1324053430 ack 3469848491
0.141726 8.1.1.50.8483 -> 11.55.113.111.80: fin 1325501489 ack 1325814854
0.142136 11.55.113.142.80 -> 8.1.1.49.37069: fin 3470735452 ack 3470140563
0.142229 8.1.1.49.37069 -> 11.55.113.142.80: fin 3470140563 ack 3470735453
0.142358 11.55.113.137.80 -> 8.1.1.37.64552: fin 3470575069 ack 3470178397
0.142470 11.55.113.50.80 -> 8.1.1.146.20570: fin 1325199483 ack 1325643886
0.142592 8.1.1.37.64552 -> 11.55.113.137.80: fin 3470178397 ack 3470575070
0.142631 8.1.1.146.20570 -> 11.55.113.50.80: fin 1325643886 ack 1325199484

LAB01-FG3600C-01 #
LAB01-FG3600C-01 #
LAB01-FG3600C-01 # diag sniffer packet port2 'tcp[13] & 4 = 4'
interfaces=[port2]
filters=[tcp[13] & 4 = 4]

0 packets received by filter
0 packets dropped by kernel

LAB01-FG3600C-01 # diag sniffer packet port2 'tcp[13] & 1 = 1'
interfaces=[port2]
filters=[tcp[13] & 1 = 1]
0.735302 8.1.1.43.45739 -> 11.55.113.113.80: fin 1325435586 ack 3476245506
0.735390 8.1.1.78.20526 -> 11.55.113.54.80: fin 3475750138 ack 3476257730
0.735402 11.55.113.118.80 -> 8.1.1.111.60733: fin 1326731313 ack 1326140831
0.735572 8.1.1.105.60736 -> 11.55.113.192.80: fin 1326140830 ack 3476268077
0.735581 11.55.113.144.80 -> 8.1.1.30.37574: fin 3476280085 ack 3476630338
0.735879 11.55.113.66.80 -> 8.1.1.110.17856: fin 3476894124 ack 3476435084
0.735885 11.55.113.173.80 -> 8.1.1.154.38693: fin 1326740069 ack 1326159762
0.735910 11.55.113.54.80 -> 8.1.1.99.58825: fin 3476902311 ack 3473130528
0.736218 8.1.1.111.60733 -> 11.55.113.118.80: fin 1326140831 ack 1326731314
0.736364 11.55.113.65.80 -> 8.1.1.75.32378: fin 3476854615 ack 3476482308
0.736372 8.1.1.30.37574 -> 11.55.113.144.80: fin 3476630338 ack 3476280086
0.736479 11.55.113.199.80 -> 8.1.1.72.22888: fin 1327309522 ack 1326892421
0.736510 8.1.1.99.58825 -> 11.55.113.54.80: fin 3473130528 ack 3476902312
0.736545 8.1.1.110.17856 -> 11.55.113.66.80: fin 3476435084 ack 3476894125
0.736549 8.1.1.154.38693 -> 11.55.113.173.80: fin 1326159762 ack 1326740070
0.736570 11.55.113.17.80 -> 8.1.1.92.8462: fin 3476849917 ack 3476425600
0.736578 11.55.113.146.80 -> 8.1.1.87.32381: fin 1327246122 ack 1326901515
0.737068 8.1.1.75.32378 -> 11.55.113.65.80: fin 3476482308 ack 3476854616
0.737177 8.1.1.72.22888 -> 11.55.113.199.80: fin 1326892421 ack 1327309523
0.737228 8.1.1.92.8462 -> 11.55.113.17.80: fin 3476425600 ack 3476849918
  
```

Copyright© 2010 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, and FortiGuard®, are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance metrics contained herein were attained in internal lab tests under ideal conditions. Network variables, different network environments and other conditions may affect performance results, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding contract with a purchaser that expressly warrants that the identified product will perform according to the performance metrics herein. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any guarantees. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable. Certain Fortinet products are licensed under U.S. Patent No. 5,623,600.